Introduction

## Introduction

Digital watermarking is the technology of inserting, and most of the time, hiding messages in noise- tolerant digital signals, such as images, videos and audios. The technology is mainly used for copyright protection but it also has applications to fields such as,

Copyright protection

Fingerprinting

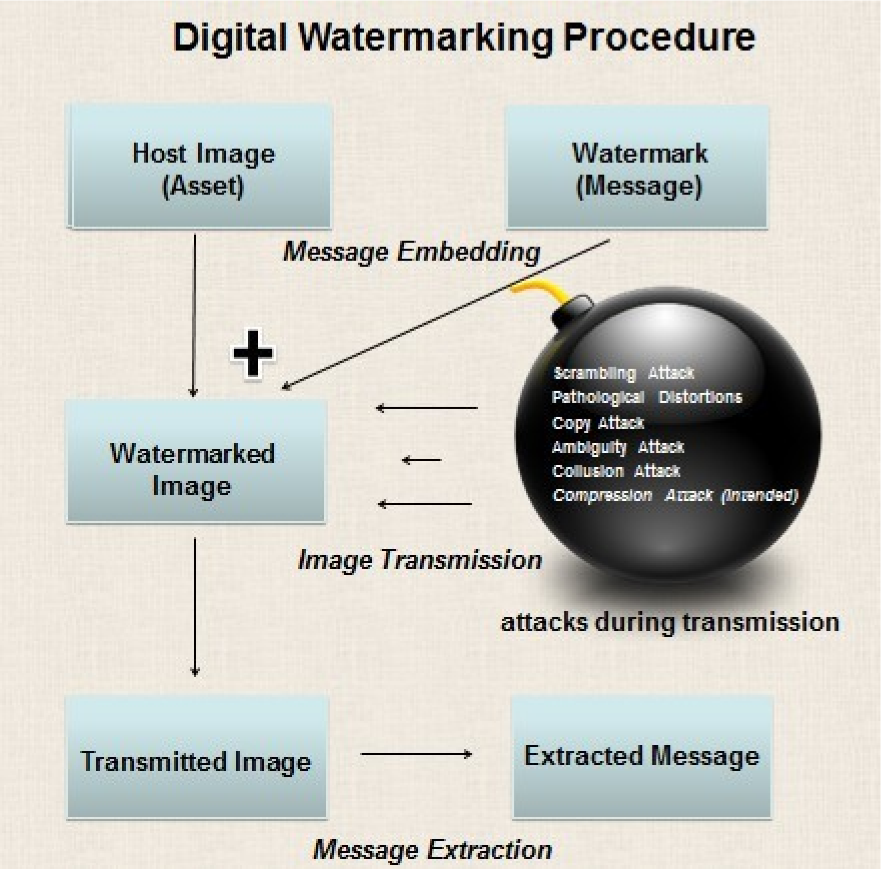
Ownership assertion

Fraud& temper detection

ID card security

Deter digital counterfeiting bank notes

The following is a top level illustration of general procedures of embedding, transmitting and extracting a digital watermark in an image.



The following is our Host and Watermarked image, which looks pretty much the same to naked eye.

Host Image

Watermarked Image

## A Key Criteria- Robustness

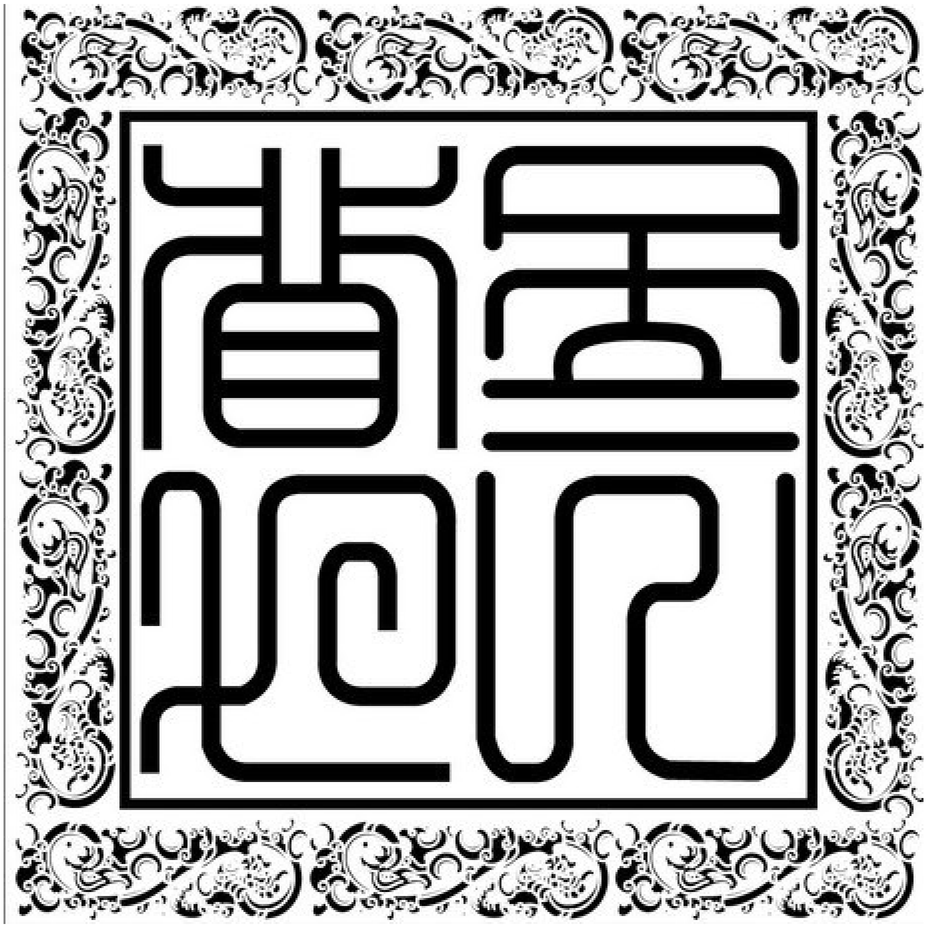
One of the major criteria of how “good” an inserted watermark is its robustness. Although fragile watermarks have occasional applications, in most cases we want to be able to extract and retrieve the original imbedded message. And this requires high robustness, which means the watermark can resist intentionally and unintentionally attacks during transmission, some of which are:

- Common signal processing such as compression, which we will go in depth with in part 4 of our report; and also analog to digital and vice versa.

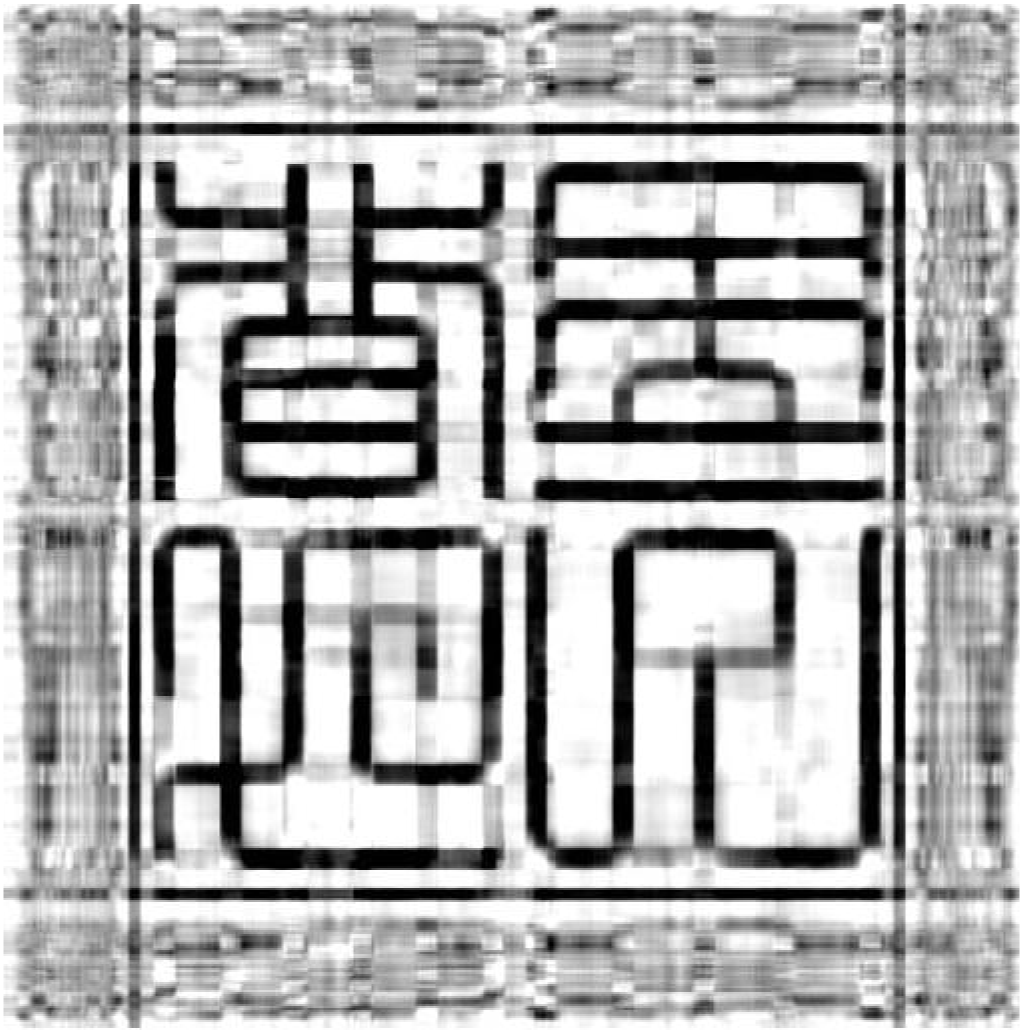
- Geometric distortion, although this has been proven hard to do.

There are several ways to measure robustness, one of which is the direct measurement of bit- wise vector. This method gives you the exact percentage of watermarked message survived. There are several other mathematic comparison methods such as Mean Square Error and Pick Signal to Noise Ratio, the latter of which includes signal strength as a factor. However, none of the above takes what similarity means for human perception into account. And as we can see towards the end of our project, two pictures can have very little number of pixels that are the same and still be recognized by human eye.

## Original Message



Extracted Message



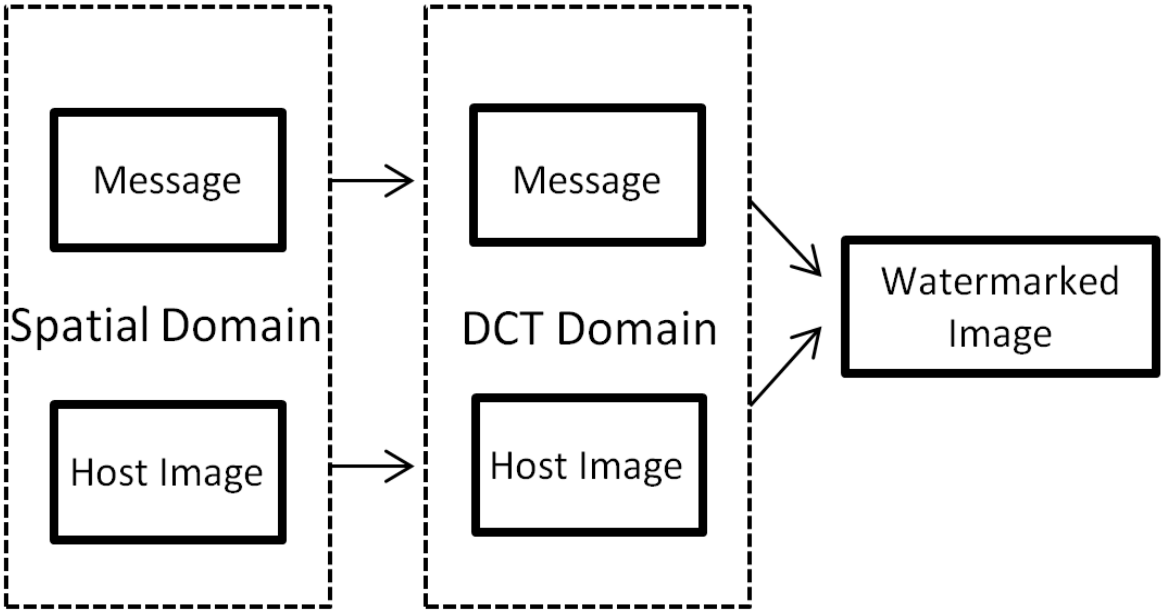
Do you believe the above pictures only have about less than 5% exact same pixels?

## Design and Implementation

Our goal is to observe and to analyze the robustness of a watermarking technique under image compression. With that in mind, it is important to choose the appropriate watermarking technique for our application. Digital watermark can be applied in the spatial domain, spectral domain, or the combination of the two with variety of different implementation methods for each. Since the focus is robustness, a spectral watermarking technique in general performs better in retaining the watermark under different attacks without significant degradation in image quality. Three popular implementations of spectral watermarking algorithms are Cox’s DCT (Discrete Cosine Transform), Xia Boncelet’s DWT (Discrete Wavelet Transform), and FM Boland’s DFT (Discrete Fourier Transoform). Cox’s algorithm has the characteristics best suited for our analysis.

Cox’s watermarking algorithm utilizes DCT to implement the watermark. Comparing to DFT, DCT yields a higher signal to noise ratio under SVD compression making it a good choice for our application. The algorithm computes the 2D DCT of the host image, image used for watermarking. To ensure the watermark can withstand attacks, the algorithm embeds the watermark in the 1000 greatest DCT values. Cox’s watermarking algorithm uses a correlation based non-blind detection system. It requires the original, unaltered message to extract the watermark. Unlike many other watermarking techniques, not only could cox’s algorithm provide an authentication check, it could also be used to send hidden messages through its watermark. Only recipients with the key, the original host image, can extract the watermarked message from the image.

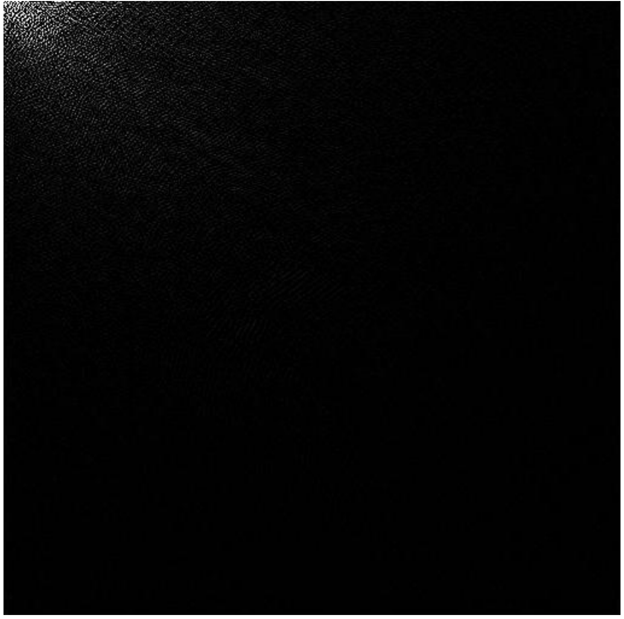
## Encoding Block Diagram



For the implementation of the watermarking system, first host image is chosen to embed the watermark. For simplicity, our analysis focuses on a 512 by 512 black and white image of the classic Lena as the host image. The watermark is represented by a string, message, with length less than or equal to 1000. The message is first feed through an encoder to convert each character into their respective ACSII representations and then scaled to between 0 and 1.

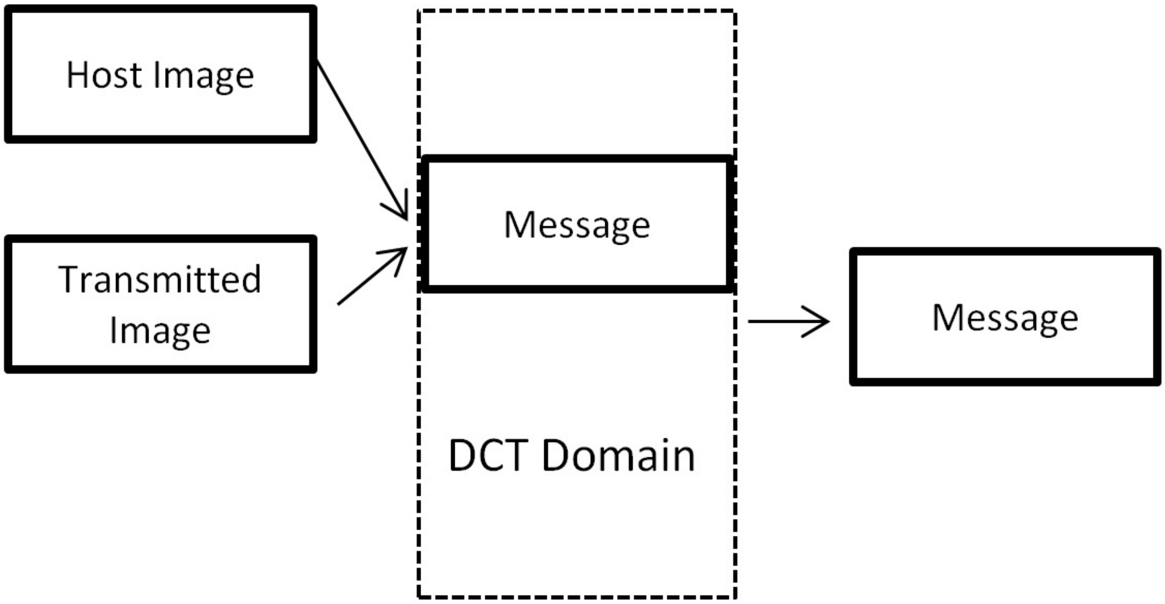
Host image

Host image's DCT representation



The resulting code is then used as the watermark. To embed the watermark to the host image, the host image is read by Matlab, converted into matrix representation, and transformed using DCT. Excluding the largest value, the DC value, the next largest 1000 DCT values are used to carry the watermark. The values multiplied by corresponding watermarking values scaled by 0.1 and then add to original values. Then the DCT matrix is converted back to spatial domain, generating a watermarked image.

Decoding Block Diagram



To extract the watermark message from the watermarked image, the reverse process is applied. The code is extracted from DCT of the watermarked image by subtracting the original value from the larger, watermarked image. The code is then converted back to the message by rescaling and matching with the characters represented by the ASCII values. Rounding is used to eliminate the rounding errors in the calculations.

## SVD Compression

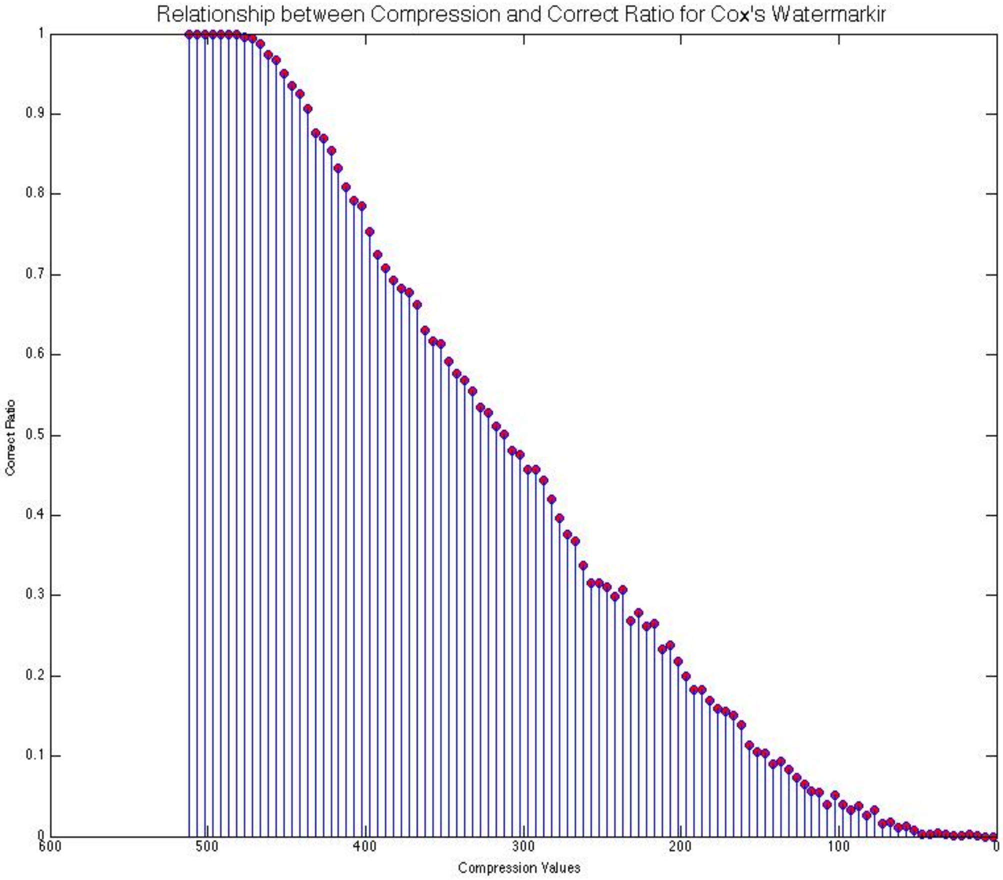
We decided to use singular value decomposition(SVD) to simulate the intentional attack and compress the image. After we embedded our message into the host image, we sent the watermarked image to the SVD compression function in Matlab. To implement SVD, we used Matlab code " [ U S V] = svd(W) " where W is the watermarked image, U,S and V are the singular values of W returned by the function.

To determine the compression ratio, we modified how many singular values we would preserve from the image. The fewer values we preserve, the larger compression ratio we have. We then wanted to extract the message from the compressed watermarked image and see if we can still retrieve the original message. We found out that the message embedded was also distorted due to the compression and the distortion was getting worse with the increase in compression ratio. Therefore, we want to test the relationship between how well the message was recovered and the compression ratio in the next step.

## Result and Analysis

To analyze the robustness of our implementation of the Cox's algorithm, a random generator function is used to generate a length-1000 string with printable characters (ACSII 32-126) for each trial of the analysis. A comparator is built to generate the ratio between number of correct character over the total number of characters by comparing the original watermark message to the extracted one. The ratio is recorded for the length-1000 randomly generated strings for each compression value starting at 512, the minimum value of the image's dimensions. To visualize the result, the correct ratio is plotted against the compression values with a decrement of 5.

At very large and small compression values, correct ratios are unresponsive to the change of compression values. Between 450-100, the correct ratio drops linearly with compression. From this result, we can conclude that effectiveness of Cox's DCT watermarking decays linearly with SVD compression at midrange and the algorithm is very effective for minor compression. Also note that error occurs for the characters with higher index before ones with lower. In context of sending secrete message through watermarking, errors are more likely to occur in longer messages than in a shorter ones under the same compression. In other words, if the message is short, one could send and retrieve the message with perfect accuracy even under high compression.


Relationship between Compression and Correct Ratio for Cox's Watermarkir
Correct Ratio
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
600
500
400
300
200
100
0
Compression Values

## Future

One of the more interesting results we found is that even with very low correct ratio, some extracted picture message are still recognizable to human perception. Therefore, based on different type of message data, we need to incorporate different evaluation methods for “correction”. One of the ways to achieve this is by using SSIM- Structure Similarity Index Method, which takes the correlation of neighboring pixels into consideration, and thus make it closer to human perception.

The other area of future development is compression method. In our project, we only implemented SVD compression. In the future we could also bring other lossy compression methods such as JPEG, which is more common in daily use.

Algorithm

1. wm_comp: compress the watermarked image at differential compression levels.file.

2. wm_comp_anal: Plotting the correct ratio, which is the percentatio of the right pixels in the extraced message, as a function of compression value.file.

3. wm_cox_ext: using cox's algorithm to extract the message inside a watermakred image.file.

4. wm_cox_image: using cox's algorithm to water mark an image.file.

5. wm_decoder: takes code words and convert it back to message.file.

6. wm_encoder: convert message (string) into a vector of acsii representation of doubles.file.

7. wm_msg_compare: analyze the similarity between the extracted message at differential compression level, with the original message.file.

8. wm_rand_msg_gen: generate a random 1000-length string as the message.file.

9. wm_runner: an top program that combines all the other modules and outputs results.file.

10.wm_sim_compress: use SVD algorithm to compress the watermarked image.file.

## The Team

Zehua Jin,

Shengrong Wu,

William Xie,

Yize Zhao

Department of Electrical & Computer Engineering, Rice University,
Houston, 77005

Posterfile.

# Reference

1.Chandramouli, R., Nasir Memon, and Majid Rabbani. "Digital watermarking." Encyclopedia of Imaging Science and Technology (2002).

2. Fridrich, Jessica, and Miroslav Goljan. "Comparing robustness of watermarking techniques." Electronic Imaging'99. International Society for Optics and Photonics, 1999.

3. Voloshynovsky, S., et al. "Generalized watermarking attack based on watermark estimation and perceptual remodulation." Proceedings of SPIE: Security and Watermarking of Multimedia Content II, San Jose, CA, USA (2000).

4. Voloshynovskiy, Sviatolsav, et al. "Attacks on digital watermarks: classification, estimation based attacks, and benchmarks." Communications Magazine, IEEE 39.8 (2001): 118-126.

5. Cox, Ingemar J., et al. "Digital watermarking." U.S. Patent No. 5,915,027. 22 Jun. 1999.

6. Cao, Lijie. "Singular value decomposition applied to digital image processing." Division of Computing Studies, Arizona State University Polytechnic Campus, Mesa, Arizona State University polytechnic Campus (2006).